CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

IT347: SOFTWARE ENGINEERING

UNIT 2

Agile Development

DEPSTAR

Content

- Agility and Agile Process Model
- Extreme Programming
- Other Process Models of Agile Development and Tools

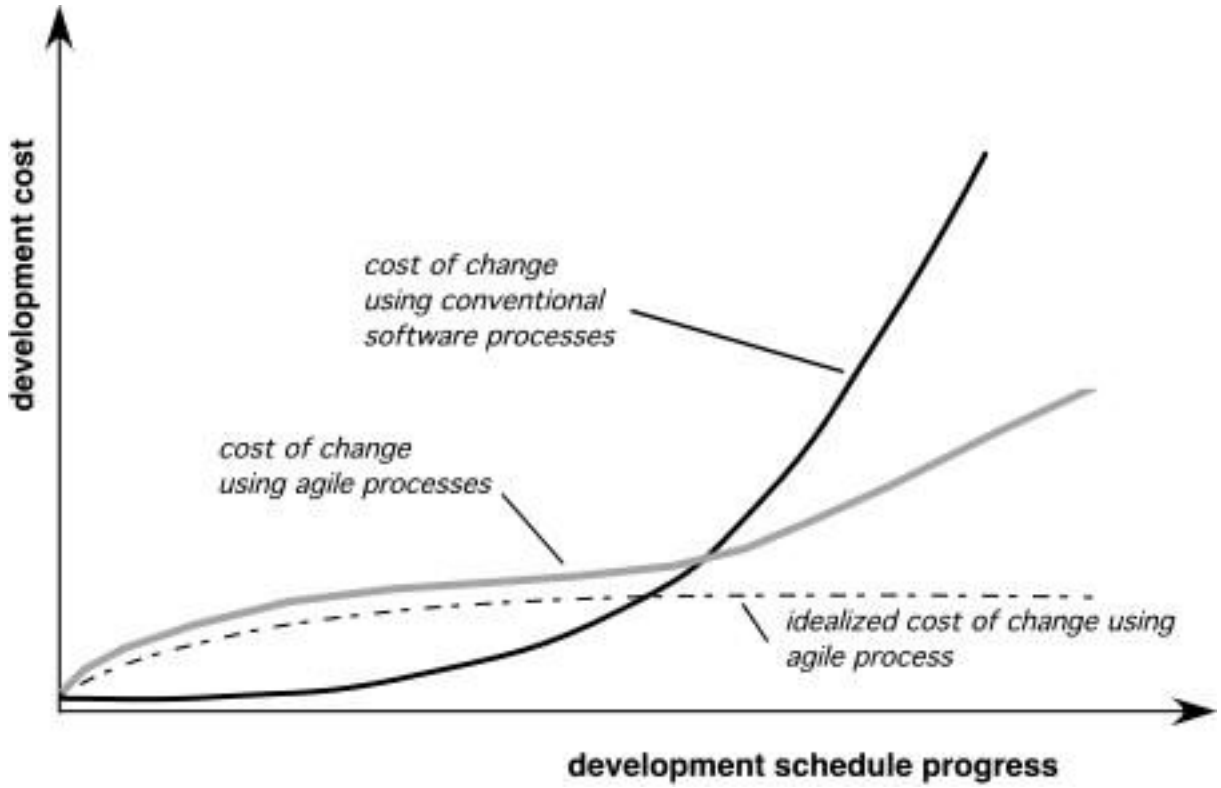
What is "Agility"?

- Effective (rapid and adaptive) response to change
- Effective communication among all stakeholders
- Drawing the customer onto the team
- Organizing a team so that it is in control of the work performed

Yielding …

- Rapid, incremental delivery of software

Agility and the Cost of Change



An Agile Process

- Is driven by customer descriptions of what is required (scenarios)
- Recognizes that plans are short-lived
- Develops software iteratively with a heavy emphasis on construction activities
- Delivers multiple 'software increments'
- Adapts as changes occur

Agility Principles - I

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face–to–face conversation.

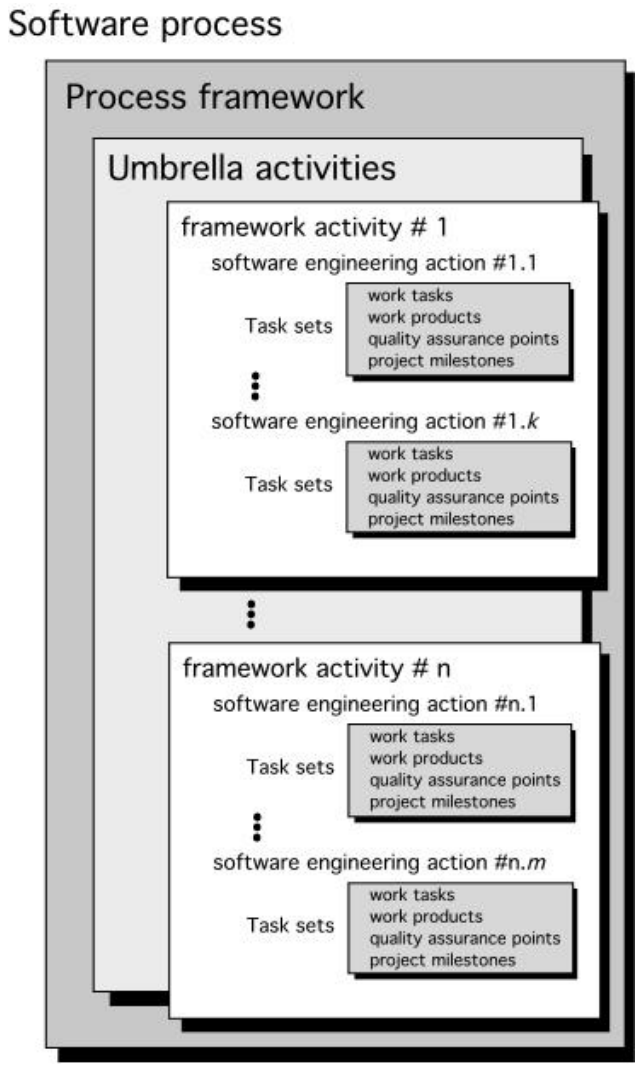
Contd…

7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity – the art of maximizing the amount of work not done – is essential.
11. The best architectures, requirements, and designs emerge from self–organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Human Factors

- *the process molds to the needs of the people and team,* not the other way around
- key traits must exist among the people on an agile team and the team itself:
  - **Competence.**
  - **Common focus.**
  - **Collaboration.**
  - **Decision-making ability.**
  - **Fuzzy problem-solving ability.**
  - **Mutual trust and respect.**
  - **Self-organization.**

Process Models



Generic Process Model

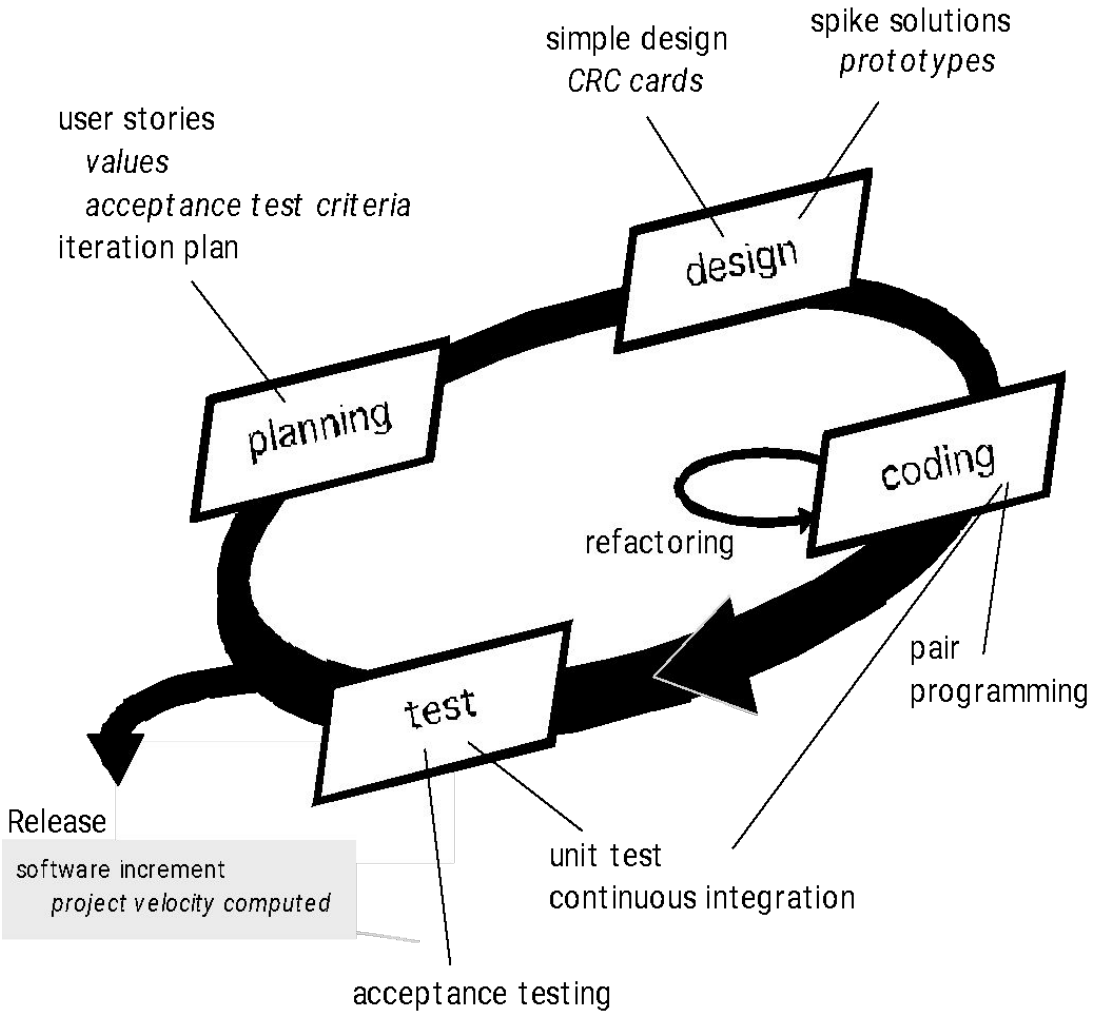
Extreme Programming (XP)

- The most widely used agile process, originally proposed by Kent Beck
- XP Planning
  - Begins with the creation of "user stories"
  - Agile team assesses each story and assigns a cost
  - Stories are grouped to for a deliverable increment
  - A commitment is made on delivery date
  - After the first increment "project velocity" is used to help define subsequent delivery dates for other increments

Contd…

- XP Design
  - Follows the KIS principle
  - Encourage the use of CRC cards (see Chapter 8)
  - For difficult design problems, suggests the creation of "spike solutions"—a design prototype
  - Encourages "refactoring"—an iterative refinement of the internal program design
- XP Coding
  - Recommends the construction of a unit test for a store *before* coding commences
  - Encourages "pair programming"
- XP Testing
  - All unit tests are executed daily
  - "Acceptance tests" are defined by the customer and executed to assess customer visible functionality

Contd…


simple design
CRC cards
spike solutions
prototypes
user stories
values
acceptance test criteria
iteration plan
design
planning
coding
refactoring
test
pair
programming
Release
software increment
project velocity computed
unit test
continuous integration
acceptance testing

Other Agile Process Models

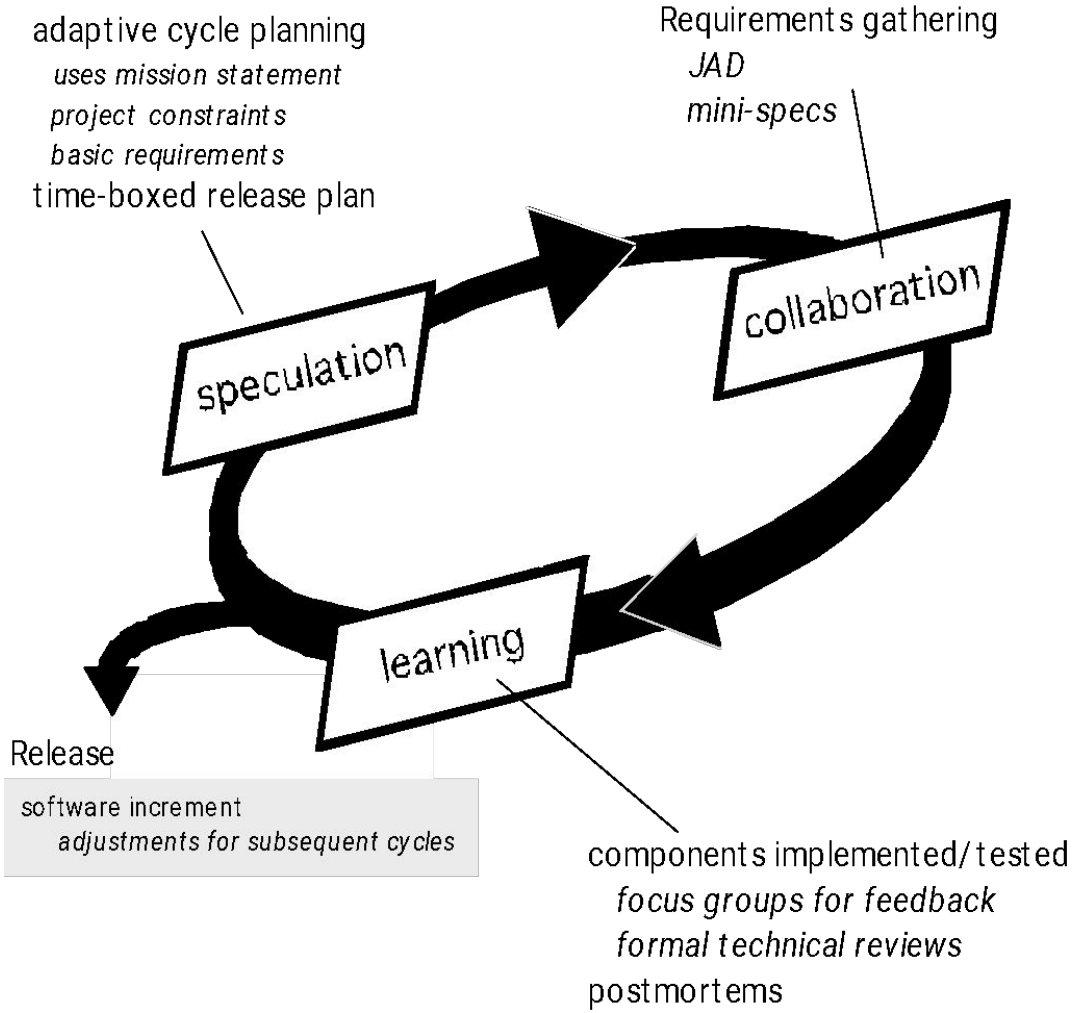
It includes:

1. Adaptive Software Development (ASD)
2. Scrum
3. Dynamic Systems Development Method (DSDM)
4. Crystal
5. Feature Driven Development (FDD)
6. Lean Software Development (LSD)
7. Agile Modeling (AM)
8. Agile Unified Process (AUP)

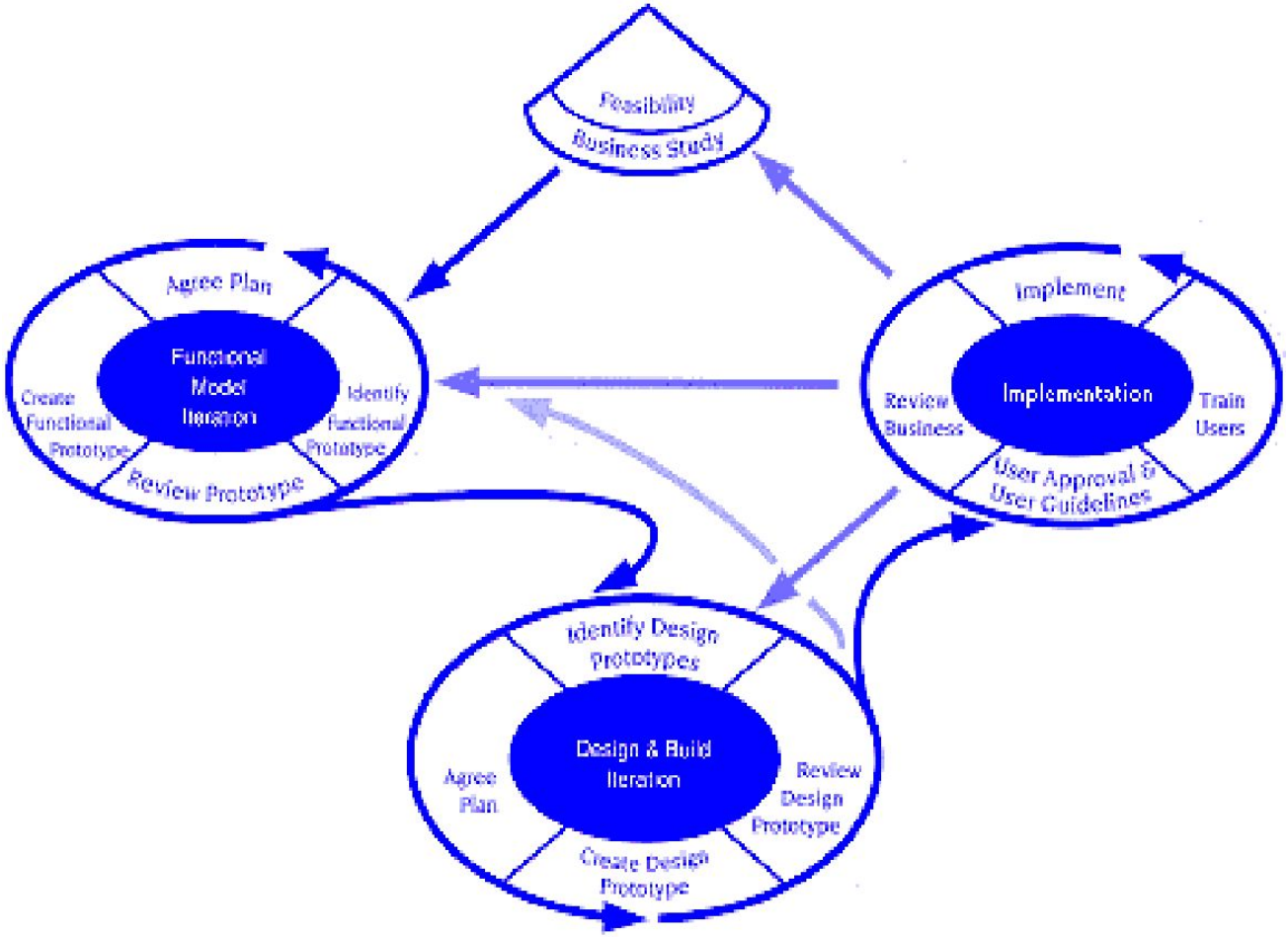
Adaptive Software Development

- Originally proposed by Jim Highsmith
- ASD — distinguishing features
  - Mission-driven planning
  - Component-based focus
  - Uses "time-boxing"
  - Explicit consideration of risks
  - Emphasizes collaboration for requirements gathering
  - Emphasizes "learning" throughout the process

Adaptive Software Development



Dynamic Systems Development Method



DSDM Life Cycle (with permission of the DSDM consortium)

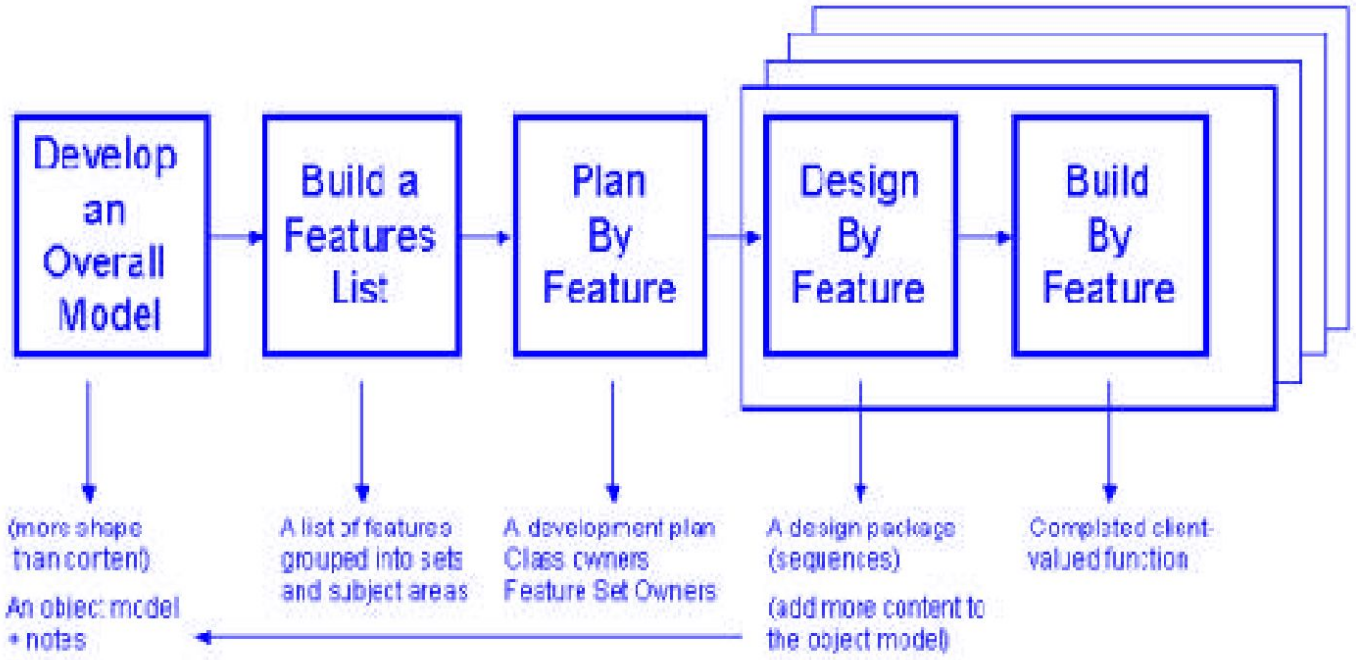
Scrum

- Originally proposed by Schwaber and Beedle
- Scrum—distinguishing features
  - Development work is partitioned into "packets"
  - Testing and documentation are on-going as the product is constructed
  - Work occurs in "sprints" and is derived from a "backlog" of existing requirements
  - Meetings are very short and sometimes conducted without chairs
  - "demos" are delivered to the customer with the time-box allocated

Crystal

- Proposed by Cockburn and Highsmith
- Crystal—distinguishing features
  - Actually a family of process models that allow "maneuverability" based on problem characteristics
  - Face-to-face communication is emphasized
  - Suggests the use of "reflection workshops" to review the work habits of the team

Feature Driven Development



Reprinted with permission of Peter Coad

Lean software development (LSD)

- It has adapted the principles of lean manufacturing to the world of software engineering.
- The lean principles can be summarized as:
  - Eliminate waste
  - Build quality in
  - Create knowledge
  - Defer commitment
  - Deliver fast
  - Respect people
  - Optimize the whole

- Each of these principles can be adapted to the software process.
- Example: eliminate the waste within the context of an agile software project can be interpreted to mean:
  - Adding no extraneous features or functions
  - Assessing the cost and schedule impact of any newly requested requirement.
  - Removing any superfluous process steps
  - Establishing the mechanisms to improve the way team members find information
  - Ensuring the testing finds as many errors as possible
  - Reducing the time required to request and get a decision that affects the software or the process that is applied to create it.
  - Streaming the manner in which information is transmitted to all stakeholders involved in the process

Agile Modeling

- Originally proposed by Scott Ambler
- Suggests a set of agile modeling principles
  - Model with a purpose
  - Use multiple models
  - Travel light
  - Content is more important than representation
  - Know the models and the tools you use to create them
  - Adapt locally

Agile Unified Process (AUP)

- It adopts a "serial in the large" and the "iterative in the small" philosophy for building computer-based systems.
- By adopting the classic UP phased activities:
  - Inception
  - Elaboration
  - Construction
  - Transition

  AUP provides a linear sequence of software engineering activities that enables the team to visualize the overall process flow for a software project.

- Each AUP iteration addresses the following activities:
  - Modeling
  - Implementation
  - Testing
  - Deployment
  - Configuration and Project Management
  - Environment Management

Tool Set for the Agile Process

- The tool set supports agile processes that focus more on people issues than it does on the technology issues.
- Active communication is achieved via the team dynamics while passive communication is achieved by "information radiators" (eg. A flat panel display that presents the overall status of different components of an increment).
- Project management tools deemphasize the Gantt Chart and replace it with earned value charts or "graphs of tests created versus passed"… other agile tools are used to optimise the environment in which the agile team works, improve the team culture by nurturing the social interactions, physical devices and process enhancement


Thank you!